

The Transcendence of Certain Infinite Series

Angeliki Koutsoukou-Argraki and Wenda Li

September 13, 2023

Abstract

We formalize the proofs of two transcendence criteria by J. Hančl and P. Rucki that assert the transcendence of the sums of certain infinite series built up by sequences that fulfil certain properties. Both proofs make use of Roth's celebrated theorem on diophantine approximations to algebraic numbers from 1955 which we implement as an assumption without having formalised its proof.

Contents

1	The transcendence of certain infinite series	1
1.1	Misc	2
1.2	Main proofs	3
1.3	Acknowledgements	23

1 The transcendence of certain infinite series

```
theory Transcendence-Series imports  
  HOL-Analysis.Multivariate-Analysis  
  HOL-Computational-Algebra.Polynomial  
  Prime-Number-Theorem.Prime-Number-Theorem-Library  
begin
```

We formalise the proofs of two transcendence criteria by J. Hančl and P. Rucki that assert the transcendence of the sums of certain infinite series built up by sequences that fulfil certain properties (Theorems 2.1 and 2.2 in [1], `HanclRucki1` and `HanclRucki2` here respectively). Both proofs make use of Roth's celebrated theorem on diophantine approximations to algebraic numbers from 1955 [2] which we assume and implement within the locale `RothsTheorem`.

A small mistake was detected in the original proof of Theorem 2.1, and the authors gave us a fix for the problem (by email). Our formalised proof incorporates this correction (see the Remark in the proof of `HanclRucki1`).

1.1 Misc

lemma *powr-less-inverse-iff*:

fixes $x\ y\ z::\text{real}$

assumes $x>0\ y>0\ z>0$

shows $x\ \text{powr}\ y < z \iff x < z\ \text{powr}\ (\text{inverse}\ y)$

proof

assume $x\ \text{powr}\ y < z$

from *powr-less-mono2*[*OF* - - *this,of inverse y*]

show $x < z\ \text{powr}\ \text{inverse}\ y$

using *assms* **by** (*auto simp:powr-powr*)

next

assume $*:x < z\ \text{powr}\ \text{inverse}\ y$

from *powr-less-mono2*[*OF* - - $*$,*of y*] **show** $x\ \text{powr}\ y < z$

using *assms* **by** (*auto simp:powr-powr*)

qed

lemma *powr-less-inverse-iff'*:

fixes $x\ y\ z::\text{real}$

assumes $x>0\ y>0\ z>0$

shows $z < x\ \text{powr}\ y \iff z\ \text{powr}\ (\text{inverse}\ y) < x$

using *powr-less-inverse-iff*[*symmetric,of - inverse y*] *assms* **by** *auto*

lemma *powr-less-eq-inverse-iff*:

fixes $x\ y\ z::\text{real}$

assumes $x>0\ y>0\ z>0$

shows $x\ \text{powr}\ y \leq z \iff x \leq z\ \text{powr}\ (\text{inverse}\ y)$

by (*meson assms not-less powr-less-inverse-iff'*)

lemma *powr-less-eq-inverse-iff'*:

fixes $x\ y\ z::\text{real}$

assumes $x>0\ y>0\ z>0$

shows $z \leq x\ \text{powr}\ y \iff z\ \text{powr}\ (\text{inverse}\ y) \leq x$

by (*simp add: assms powr-less-eq-inverse-iff*)

lemma *tendsto-PInfty-mono*:

assumes $(\text{ereal}\ o\ f) \longrightarrow \infty\ \forall_F\ x\ \text{in}\ \text{sequentially.}\ f\ x \leq g\ x$

shows $(\text{ereal}\ o\ g) \longrightarrow \infty$

using *assms* **unfolding** *comp-def tendsto-PInfty-eq-at-top*

by (*elim filterlim-at-top-mono, simp*)

lemma *limsup-infinity-imp-Inf-many*:

assumes $\text{limsup}\ f = \infty$

shows $(\forall\ m. (\exists\ \infty\ i. f\ i > \text{ereal}\ m))$ **unfolding** *INFM-nat*

proof (*clarify,rule ccontr*)

fix $m\ k$ **assume** $\neg (\exists\ n>k. \text{ereal}\ m < f\ n)$

then have $\forall\ n>k. f\ n \leq \text{ereal}\ m$ **by** *auto*

then have $\forall_F\ n\ \text{in}\ \text{sequentially.}\ f\ n \leq \text{ereal}\ m$

using *eventually-at-top-dense* **by** *blast*

then have $\text{limsup}\ f \leq \text{ereal}\ m$ **using** *Limsup-bounded* **by** *auto*

then show *False* using *assms* by *simp*
qed

lemma *snd-quotient-plus-leq*:

defines $de \equiv (snd \ o \ quotient-of)$

shows $de \ (x+y) \leq de \ x * de \ y$

proof –

obtain $x1 \ x2 \ y1 \ y2$ where $xy: \ quotient-of \ x = (x1, x2) \ \ quotient-of \ y = (y1, y2)$

by (*meson surj-pair*)

have $x2 > 0 \ y2 > 0$ using $xy \ \ quotient-of-denom-pos$ by *blast+*

then show *?thesis*

unfolding *de-def comp-def rat-plus-code xy*

apply (*auto split:prod.split simp:Rat.normalize-def Let-def*)

by (*smt div-by-1 gcd-pos-int int-div-less-self mult-eq-0-iff mult-sign-intros(1)*)

qed

lemma *quotient-of-inj*: *inj quotient-of*

unfolding *inj-def* by (*simp add: quotient-of-inject*)

lemma *infinite-inj-imageE*:

assumes *infinite A inj-on f A f ' A* $\subseteq B$

shows *infinite B*

using *assms inj-on-finite* by *blast*

lemma *incseq-tendsto-limsup*:

fixes $f :: nat \Rightarrow 'a :: \{complete-linorder, linorder-topology\}$

assumes *incseq f*

shows $f \longrightarrow \limsup \ f$

using *LIMSEQ-SUP assms convergent-def convergent-ereal tendsto-Limsup*

trivial-limit-sequentially by *blast*

1.2 Main proofs

Since the proof of Roth's theorem has not been formalized yet, we formalize the statement in a locale and use it as an assumption.

locale *RothsTheorem* =

assumes *RothsTheorem*: $\forall \xi \ \kappa. \ algebraic \ \xi \wedge \xi \notin \mathbf{Q} \wedge \ infinite \ \{(p, q). \ q > 0 \wedge \ coprime \ p \ q \wedge |\xi - of-int \ p / of-int \ q| < 1 / q \ powr \ \kappa\} \longrightarrow \kappa \leq 2$

theorem (in *RothsTheorem*) *HanclRucki1*:

fixes $a \ b :: nat \Rightarrow int$ and $\delta :: real$

defines $aa \equiv (\lambda n. \ real-of-int \ (a \ n))$ and $bb \equiv (\lambda n. \ real-of-int \ (b \ n))$

assumes *a-pos*: $\forall k. \ a \ k > 0$ and *b-pos*: $\forall k. \ b \ k > 0$ and $\delta > 0$

and $\limsup\text{-infy}:\limsup \ (\lambda k. \ aa \ (k+1) / (\prod_{i=0..k} \ aa \ i) \ powr \ (2+\delta) * (1 / bb \ (k+1))) = \infty$

and $\liminf\text{-1}:\liminf \ (\lambda k. \ aa \ (k+1) / aa \ k * bb \ k / bb \ (k+1)) > 1$

shows $\neg \ algebraic(\suminf \ (\lambda k. \ bb \ k / aa \ k))$

proof –

have *summable:summable* $(\lambda k. \ bb \ k / aa \ k)$

proof (*rule ratio-test-convergence*)
have [*simp*]: $aa\ k > 0\ bb\ k > 0$ **for** k
unfolding *aa-def bb-def* **using** *a-pos b-pos* **by** *auto*
show $\forall_F n$ *in sequentially. $0 < bb\ n / aa\ n$*
by *auto*
show $1 < \liminf (\lambda n. \text{ereal } (bb\ n / aa\ n / (bb\ (Suc\ n) / aa\ (Suc\ n))))$
using *liminf-1* **by** (*auto simp: algebra-simps*)
qed
have [*simp*]: $aa\ k > 0\ bb\ k > 0$ **for** k **unfolding** *aa-def bb-def*
by (*auto simp add: a-pos b-pos*)
have *ab-1*: $aa\ k \geq 1\ bb\ k \geq 1$ **for** k
unfolding *aa-def bb-def* **using** *a-pos b-pos*
by (*auto simp add: int-one-le-iff-zero-less*)

define B **where** $B \equiv \liminf (\lambda x. \text{ereal } (aa\ (x + 1) / aa\ x * bb\ x / bb\ (x + 1)))$
define M **where** $M \equiv (\text{case } B \text{ of } \text{ereal } m \Rightarrow (m+1)/2 \mid - \Rightarrow 2)$
have $M > 1\ M < B$
using *liminf-1* **unfolding** *M-def*
by (*auto simp add: M-def B-def split: ereal.split*)

Remark: In the original proof of Theorem 2.1 in [1] it was claimed in p.534 that from assumption (3) (i.e. $1 < \liminf (\lambda x. \text{ereal } (aa\ (x + 1) / aa\ x * bb\ x / bb\ (x + 1)))$), we obtain that: $\forall A > 1 \exists k_0 \forall k > k_0 \frac{1}{A} \frac{b_k}{a_k} > \frac{b_{k+1}}{a_{k+1}}$, however note the counterexample where $a_{k+1} = k(a_1 a_2 \dots a_k)^{\lceil 2+\delta \rceil}$ if k is odd, and $a_{k+1} = 2a_k$ otherwise, with $b_k = 1$ for all k . In communication by email the authors suggested to replace the claim $\forall A > 1 \exists k_0 \forall k > k_0 \frac{1}{A} \frac{b_k}{a_k} > \frac{b_{k+1}}{a_{k+1}}$ with $\exists A > 1 \exists k_0 \forall k > k_0 \frac{1}{A} \frac{b_k}{a_k} > \frac{b_{k+1}}{a_{k+1}}$ which solves the problem and the proof proceeds as in the paper. The witness for $\exists A > 1$ is denoted by M here.

have *bb-aa-event*: $\forall_F k$ *in sequentially. $(1/M) * (bb\ k / aa\ k) > bb\ (k+1) / aa\ (k+1)$*
using *less-LiminfD[OF <M < B>[unfolding B-def],simplified]*
by *eventually-elim (use <M > 1> in <auto simp: field-simps>)*

have *bb-aa-p*: $\forall_F k$ *in sequentially. $\forall p. bb\ (k+p) / aa\ (k+p) \leq (1/M \hat{p}) * (bb\ k / aa\ k)$*

proof –

obtain k_0 **where** *k0-ineq*:

$\forall n \geq k_0. bb\ (n + 1) / aa\ (n + 1) < 1 / M * (bb\ n / aa\ n)$

using *bb-aa-event* **unfolding** *eventually-sequentially*

by *auto*

have $bb\ (k+p) / aa\ (k+p) \leq (1/M \hat{p}) * (bb\ k / aa\ k)$ **when** $k \geq k_0$ **for** $p\ k$

proof (*induct p*)

case 0

then show *?case* **by** *auto*

next

case (*Suc p*)

have $bb\ (k + Suc\ p) / aa\ (k + Suc\ p) < 1 / M * (bb\ (k+p) / aa\ (k+p))$

```

    using k0-ineq[rule-format, of k+p] that by auto
  also have ... ≤ 1 / M ^ (Suc p) * (bb k / aa k)
    using Suc ⟨M>1⟩ by (auto simp add:field-simps)
  finally show ?case by auto
qed
then show ?thesis unfolding eventually-sequentially by auto
qed

define ξ where ξ = suminf (λ k. bb k / aa k)
have ξ-Inf-many: ∃ ∞ k. |ξ - (∑ k = 0..k. bb k / aa k)| < 1 / prod aa {0..k}
powr (2 + δ)
proof -
  have |ξ - (∑ i = 0..k. bb i / aa i)| = |∑ i. bb (i+(k+1)) / aa (i+(k+1))|
for k
  unfolding ξ-def
  apply (subst suminf-minus-initial-segment[of - k+1, OF summable])
  using atLeast0AtMost lessThan-Suc-atMost by auto

moreover have ∃ ∞ k. |∑ i. bb(i+(k+1)) / aa (i+(k+1))|
< 1 / prod aa {0..k} powr (2 + δ)
proof -
  define P where P ≡ (λ i. ∃ p. bb (i + 1 + p) / aa (i + 1 + p)
≤ 1 / M ^ p * (bb (i + 1) / aa (i + 1)))
  define Q where Q ≡ (λ i. ereal (M / (M - 1))
< ereal (aa (i + 1) / prod aa {0..i} powr (2 + δ)) * (1 / bb (i +
1))))
  have ∃ ∞ i. P i
    using bb-aa-p[THEN sequentially-offset, of 1] cofinite-eq-sequentially
    unfolding P-def by auto
  moreover have ∃ ∞ i. Q i
    using limsup-infy[THEN limsup-infinity-imp-Inf-many, rule-format, of (M /
(M - 1))]
    unfolding Q-def .
  moreover have |∑ i. bb(i+(k+1)) / aa (i+(k+1))|
< 1 / prod aa {0..k} powr (2 + δ)
  when P k Q k for k
proof -
  have summable-M: summable (λ i. 1 / M ^ i)
    apply (rule summable-ratio-test[of 1/M])
    using ⟨M>1⟩ by auto

  have (∑ i. bb (i + (k + 1)) / aa (i + (k + 1))) ≥ 0
    apply (rule suminf-nonneg)
  subgoal using summable-ignore-initial-segment[OF summable, of k+1] by
auto
  subgoal by (simp add: less-imp-le)
  done
  then have |∑ i. bb (i + (k + 1)) / aa (i + (k + 1))|
= (∑ i. bb (i + (k + 1)) / aa (i + (k + 1)))

```

```

    by auto
  also have ... ≤ (∑ i. 1 / M ^ i * (bb (k + 1) / aa (k + 1)))
    apply (rule suminf-le)
  subgoal using that(1) unfolding P-def by (auto simp add: algebra-simps)
  subgoal using summable-ignore-initial-segment[OF summable, of k+1] by
auto
    subgoal using summable-mult2[OF summable-M, of bb (k + 1) / aa (k
+ 1)]
      by auto
    done
  also have ... = (bb (k + 1) / aa (k + 1)) * (∑ i. 1 / M ^ i)
    using suminf-mult2[OF summable-M, of bb (k + 1) / aa (k + 1)]
    by (auto simp: algebra-simps)
  also have ... = (bb (k + 1) / aa (k + 1)) * (∑ i. (1 / M) ^ i)
    using ⟨M>1⟩ by (auto simp: field-simps)
  also have ... = (bb (k + 1) / aa (k + 1)) * (M / (M - 1))
    apply (subst suminf-geometric)
    using ⟨M>1⟩ by (auto simp: field-simps)
  also have ... < (bb (k + 1) / aa (k + 1)) * (aa (k + 1) /
    prod aa {0..k} powr (2 + δ) * (1 / bb (k + 1)))
    apply (subst mult-less-cancel-left-pos)
    using that(2) unfolding Q-def by auto
  also have ... = 1 / prod aa {0..k} powr (2 + δ)
    using ab-1[of Suc k] by auto
  finally show ?thesis .
qed
ultimately show ?thesis by (smt INFM-conjI INFM-mono)
qed
ultimately show ?thesis by auto
qed

define pq where pq ≡ (λk. quotient-of (∑ i=0..k. of-int (b i) / of-int (a i)))
define p q where p ≡ fst o pq and q = snd o pq
have coprime-pq: coprime (p k) (q k)
  and q-pos: q k > 0 and pq-sum: p k / q k = (∑ i=0..k. b i / a i) for k
proof -
  have eq: quotient-of (∑ i=0..k. of-int (b i) / of-int (a i)) = (p k, q k)
    by (simp add: p-def q-def pq-def)
  from quotient-of-coprime[OF eq] show coprime (p k) (q k) .
  from quotient-of-denom-pos[OF eq] show q k > 0 .
  have (∑ i=0..k. b i / a i) = of-rat (∑ i=0..k. of-int (b i) / of-int (a i))
    by (simp add: of-rat-sum of-rat-divide)
  also have (∑ i=0..k. rat-of-int (b i) / rat-of-int (a i)) =
    rat-of-int (p k) / rat-of-int (q k)
    using quotient-of-div[OF eq] by simp
  finally show p k / q k = (∑ i=0..k. b i / a i) by (simp add: of-rat-divide)
qed

have ξ-Inf-many2: ∃∞ k. |ξ - p k / q k| < 1 / q k powr (2 + δ)

```

```

using  $\xi$ -Inf-many
proof (elim INFM-mono)
  fix  $k$  assume  $asm: |\xi - (\sum k = 0..k. bb\ k / aa\ k)| < 1 / \text{prod } aa\ \{0..k\} \text{ powr } (2 + \delta)$ 
  have  $|\xi - \text{real-of-int } (p\ k) / \text{real-of-int } (q\ k)|$ 
     $= |\xi - (\sum k = 0..k. bb\ k / aa\ k)|$ 
    using  $pq$ -sum unfolding  $aa$ -def  $bb$ -def by auto
  also have  $\dots < 1 / \text{prod } aa\ \{0..k\} \text{ powr } (2 + \delta)$ 
    using  $asm$  by auto
  also have  $\dots \leq 1 / q\ k \text{ powr } (2 + \delta)$ 
proof -
  have  $q\ k \leq \text{prod } aa\ \{0..k\}$ 
proof (induct  $k$ )
  case 0
  then show ?case unfolding  $q$ -def  $pq$ -def  $aa$ -def
    apply (simp add:rat-divide-code of-int-rat quotient-of-Fract)
    using  $ab-1$ [of 0,unfolded  $aa$ -def  $bb$ -def] unfolding  $Let$ -def  $normalize$ -def
    apply auto
  by (metis div-by-1 gcd-pos-int less-imp-le less-trans nonneg1-imp-zdiv-pos-iff
    not-less zdiv-mono2)
next
case (Suc  $k$ )
define  $de$  where  $de \equiv snd \circ \text{quotient-of}$ 
have  $\text{real-of-int } (q\ (Suc\ k))$ 
   $= de\ (\sum i=0..Suc\ k. \text{of-int } (b\ i) / \text{of-int } (a\ i))$ 
  unfolding  $q$ -def  $pq$ -def  $de$ -def by simp
also have  $\dots = de\ ((\sum i=0..k. \text{of-int } (b\ i) / \text{of-int } (a\ i))$ 
   $+ \text{of-int } (b\ (Suc\ k)) / \text{of-int } (a\ (Suc\ k)))$ 
  by simp
also have  $\dots \leq de\ (\sum i=0..k. \text{of-int } (b\ i) / \text{of-int } (a\ i))$ 
   $* de\ (\text{of-int } (b\ (Suc\ k)) / \text{of-int } (a\ (Suc\ k)))$ 
  using  $snd$ -quotient-plus-leq[folded  $de$ -def] by presburger
also have  $\dots = q\ k * de\ (\text{of-int } (b\ (Suc\ k)) / \text{of-int } (a\ (Suc\ k)))$ 
  unfolding  $q$ -def  $pq$ -def  $de$ -def by auto
also have  $\dots = q\ k * snd\ (Rat.normalize\ (b\ (Suc\ k), a\ (Suc\ k)))$ 
  by (simp add:rat-divide-code of-int-rat quotient-of-Fract  $de$ -def)
also have  $\dots \leq q\ k * aa\ (Suc\ k)$ 
  using  $ab-1$ [of  $Suc\ k$ ]  $q$ -pos[of  $k$ ]
  unfolding  $normalize$ -def  $aa$ -def  $bb$ -def  $Let$ -def
  apply auto
  by (metis div-by-1 int-one-le-iff-zero-less less-trans
    nonneg1-imp-zdiv-pos-iff not-less zdiv-mono2 zero-less-one)
also have  $\dots \leq \text{prod } aa\ \{0..k\} * aa\ (Suc\ k)$ 
  using  $Suc\ ab-1$ [of  $Suc\ k$ ] by auto
also have  $\dots = \text{prod } aa\ \{0..Suc\ k\}$ 
  by (simp add: prod.atLeast0-atMost-Suc)
finally show ?case .
qed

```

```

then show ?thesis
  by (smt ⟨0 < δ⟩ frac-le of-int-0 of-int-le-iff powr-gt-zero
        powr-mono2 q-pos)
qed
finally show |ξ - real-of-int (p k) / real-of-int (q k)| < 1 / real-of-int (q k)
powr (2 + δ) .
qed

define pqs where pqs ≡ {(p, q). q > 0 ∧ coprime p q
  ∧ |ξ - real-of-int p / real-of-int q| < 1 / q powr (2 + δ)}
have ξ-infinite: infinite pqs
proof -
  define A where A ≡ {k. |ξ - (p k) / (q k)| < 1 / (q k) powr (2 + δ)}
  have ∃∞ k. |ξ - p k / q k| < 1 / q k powr (2 + δ)
    using ξ-Inf-many2 .
  then have infinite A
    unfolding Inf-many-def A-def by auto
  moreover have inj-on (λk. (p k, q k)) A
proof -
  define g where g ≡ (λi. rat-of-int (b i) / rat-of-int (a i))
  define f where f ≡ (λk. ∑ i = 0..k. g i)
  have g-pos: g i > 0 for i
    unfolding g-def by (simp add: a-pos b-pos)
  have strict-mono f unfolding strict-mono-def f-def
proof safe
  fix x y::nat assume x < y
  then have sum g {0..y} - sum g {0..x} = sum g {x<..y}
    apply (subst Groups-Big.sum-diff[symmetric])
    by (auto intro: arg-cong2[where f=sum])
  also have ... > 0
    apply (rule ordered-comm-monoid-add-class.sum-pos)
    using ⟨x < y⟩ g-pos by auto
  finally have sum g {0..y} - sum g {0..x} > 0 .
  then show sum g {0..x} < sum g {0..y} by auto
qed
then have inj f using strict-mono-imp-inj-on by auto
then have inj (quotient-of o f) by (simp add: inj-compose quotient-of-inj)
then have inj (λk. (p k, q k))
  unfolding f-def p-def q-def pq-def comp-def
  apply (fold g-def)
  by auto
then show ?thesis by (auto elim: subset-inj-on)
qed
moreover have (λk. (p k, q k)) ‘ A ⊆ pqs
  unfolding A-def pqs-def using coprime-pq q-pos by auto
ultimately show ?thesis
  apply (elim infinite-inj-imageE)
  by auto
qed

```


moreover have *finite pqs* **if** $\xi \in \mathbb{Q}$
proof –
obtain $m\ n$ **where** $\xi\text{-mn}:\xi = (\text{of-int } m / \text{of-int } n)$ **and** *coprime m n n > 0*
proof –
obtain $m\ n$ **where** $\text{mn}:\xi = (\text{of-nat } m / \text{of-nat } n)$ *coprime m n n ≠ 0*
using *Rats-abs-nat-div-natE*[*OF* $\langle \xi \in \mathbb{Q} \rangle$ *Rats-abs-nat-div-natE*]
by *metis*
define m' **and** $n':\text{int}$
where $m' = (\text{if } \xi > 0 \text{ then nat } m \text{ else } -\text{nat } m)$ **and** $n' = \text{nat } n$
then have $\xi = (\text{of-int } m' / \text{of-int } n')$ *coprime m' n' n' > 0*
using mn **by** *auto*
then show *?thesis* **using** *that by auto*
qed
have $pqs \subseteq \{(m,n)\} \cup \{x. x \in pqs \wedge -|m| - 1 \leq \text{fst } x \wedge \text{fst } x \leq |m| + 1 \wedge 0 < \text{snd } x \wedge \text{snd } x < n\}$
proof (*rule subsetI*)
fix x **assume** $x \in pqs$
define $p\ q$ **where** $p \equiv \text{fst } x$ **and** $q = \text{snd } x$
have $q > 0$ *coprime p q* **and** $\text{pq-less}:\xi - p / q < 1 / q \text{ powr } (2 + \delta)$
using $\langle x \in pqs \rangle$ **unfolding** *p-def q-def pqs-def* **by** *auto*
have $q\text{-lt-}n:q < n$ **when** $m \neq p \vee n \neq q$
proof –
have $m * q \neq n * p$ **using** *that* $\langle \text{coprime } m\ n \rangle \langle \text{coprime } p\ q \rangle \langle q > 0 \rangle \langle n > 0 \rangle$
by (*metis eq-rat(1) fst-conv int-one-le-iff-zero-less mult.commute normalize-stable*
not-one-le-zero quotient-of-Fract snd-conv)
then have $1 / (n * q) \leq |m/n - p/q|$
using $\langle q > 0 \rangle \langle n > 0 \rangle$
apply (*auto simp:field-simps*)
by (*metis add-diff-cancel-left' diff-diff-eq2 diff-zero less-irrefl not-le of-int-diff*
of-int-lessD of-int-mult)
also have $\dots < 1 / q \text{ powr } (2 + \delta)$
using pq-less **unfolding** $\xi\text{-mn}$ **by** *auto*
also have $\dots \leq 1 / q^2$
proof –
have $\text{real-of-int } (q^2) = q \text{ powr } 2$
apply (*subst powr-numeral*)
unfolding *power2-eq-square* **using** $\langle q > 0 \rangle$ **by** *auto*
also have $\dots \leq q \text{ powr } (2 + \delta)$
apply (*rule powr-mono*)
using $\langle q > 0 \rangle \langle \delta > 0 \rangle$ **by** *auto*
finally have $\text{real-of-int } (q^2) \leq \text{real-of-int } q \text{ powr } (2 + \delta)$.
moreover have $\text{real-of-int } q \text{ powr } (2 + \delta) > 0$ **using** $\langle 0 < q \rangle$ **by** *auto*
ultimately show *?thesis* **by** (*auto simp:field-simps*)
qed
finally have $1 / (n * q) < 1 / q^2$.
then show *?thesis* **using** $\langle q > 0 \rangle \langle n > 0 \rangle$
unfolding *power2-eq-square* **by** (*auto simp:field-simps*)

```

qed
moreover have  $- |m| - 1 \leq p \wedge p \leq |m| + 1$  when  $m \neq p \vee n \neq q$ 
proof -
  define  $qn$  where  $qn \equiv q/n$ 
  have  $0 < qn \ qn < 1$  unfolding  $qn-def$  using  $q-lt-n[OF \langle m \neq p \vee n \neq q \rangle] \langle q > 0 \rangle$ 
by auto

  have  $|m/n - p/q| < 1/q \text{ powr } (2 + \delta)$  using  $pq-less$  unfolding  $\xi-mn$ 
by simp
  then have  $|p/q - m/n| < 1/q \text{ powr } (2 + \delta)$  by simp
  then have  $m/n - 1/q \text{ powr } (2 + \delta) < p/q \wedge p/q < m/n + 1/q \text{ powr } (2 + \delta)$ 
  unfolding  $abs-diff-less-iff$  by auto
  then have  $qn*m - q/q \text{ powr } (2 + \delta) < p \wedge p < qn*m + q/q \text{ powr } (2 + \delta)$ 
  unfolding  $qn-def$  using  $\langle q > 0 \rangle$  by (auto simp:field-simps)
  moreover have  $- |m| - 1 \leq qn*m - q/q \text{ powr } (2 + \delta)$ 
  proof -
    have  $- |m| \leq qn*m$  using  $\langle 0 < qn \rangle \langle qn < 1 \rangle$ 
    apply (cases m ≥ 0)
    subgoal
      apply simp
    by (meson less-eq-real-def mult-nonneg-nonneg neg-le-0-iff-le of-int-0-le-iff order-trans)
    subgoal by simp
    done
  moreover have  $- 1 \leq - q/q \text{ powr } (2 + \delta)$ 
  proof -
    have  $q = q \text{ powr } 1$  using  $\langle 0 < q \rangle$  by auto
    also have  $\dots \leq q \text{ powr } (2 + \delta)$ 
    apply (rule powr-mono)
    using  $\langle q > 0 \rangle \langle \delta > 0 \rangle$  by auto
    finally have  $q \leq q \text{ powr } (2 + \delta)$  .
    then show ?thesis using  $\langle 0 < q \rangle$  by auto
  qed
  ultimately show ?thesis by auto
qed
moreover have  $qn*m + q/q \text{ powr } (2 + \delta) \leq |m| + 1$ 
proof -
  have  $qn*m \leq |m|$  using  $\langle 0 < qn \rangle \langle qn < 1 \rangle$ 
  apply (cases m ≥ 0)
  subgoal by (simp add: mult-left-le-one-le)
  subgoal by (smt of-int-0-le-iff zero-le-mult-iff)
  done
  moreover have  $q/q \text{ powr } (2 + \delta) \leq 1$ 
  proof -
    have  $q = q \text{ powr } 1$  using  $\langle 0 < q \rangle$  by auto
    also have  $\dots \leq q \text{ powr } (2 + \delta)$ 
    apply (rule powr-mono)

```

using $\langle q > 0 \rangle \langle \delta > 0 \rangle$ by auto
 finally have $q \leq q \text{ powr } (2 + \delta)$.
 then show ?thesis using $\langle 0 < q \rangle$ by auto
 qed
 ultimately show ?thesis by auto
 qed
 ultimately show ?thesis by auto
 qed
 ultimately show $x \in \{(m, n)\} \cup \{x \in pqs. - |m| - 1 \leq fst\ x \wedge fst\ x \leq |m|$
 $+ 1$
 $\wedge 0 < snd\ x \wedge snd\ x < n\}$
 using $\langle x \in pqs \rangle \langle q > 0 \rangle$ unfolding p-def q-def by force
 qed
 moreover have finite $\{x. x \in pqs \wedge - |m| - 1 \leq fst\ x \wedge fst\ x \leq |m| + 1 \wedge$
 $0 < snd\ x \wedge snd\ x < n\}$
 proof -
 have finite $(\{- |m| - 1..|m| + 1\} \times \{0 <..<n\})$ by blast
 moreover have $\{x. x \in pqs \wedge - |m| - 1 \leq fst\ x \wedge fst\ x \leq |m| + 1 \wedge 0 < snd$
 $x \wedge snd\ x < n\} \subseteq$
 $(\{- |m| - 1..|m| + 1\} \times \{0 <..<n\})$
 by auto
 ultimately show ?thesis
 apply (elim rev-finite-subset)
 by blast
 qed
 ultimately show ?thesis using finite-subset by auto
 qed
 ultimately show ?thesis
 apply (fold ξ -def)
 using RothsTheorem[rule-format, of $\xi\ 2 + \delta$, folded pqs-def] $\langle \delta > 0 \rangle$ by auto
 qed

theorem (in RothsTheorem) HanclRucki2:

fixes $a\ b :: nat \Rightarrow int$ and $\delta\ \varepsilon :: real$
 defines $aa \equiv (\lambda n. real-of-int\ (a\ n))$ and $bb \equiv (\lambda n. real-of-int\ (b\ n))$
 assumes a-pos: $\forall k. a\ k > 0$ and b-pos: $\forall k. b\ k > 0$ and $\delta > 0$
 and $\varepsilon > 0$
 and $limsup\text{-infi} : limsup\ (\lambda k. (aa\ (k+1) / (\prod i = 0..k. aa\ i) \text{ powr } (2 + (2/\varepsilon) + \delta))$
 $* (1 / (bb\ (k+1)))) = \infty$
 and ratio-large: $\forall k. (k \geq t \longrightarrow ((aa\ (k+1) / bb\ (k+1)) \text{ powr } (1 / (1 + \varepsilon)))$
 $\geq ((aa\ k / bb\ k) \text{ powr } (1 / (1 + \varepsilon))) + 1)$
 shows $\neg algebraic (suminf\ (\lambda k. bb\ k / aa\ k))$
 proof -
 have aa-bb-pos[simp]: $aa\ k > 0\ bb\ k > 0$ for k
 unfolding aa-def bb-def using a-pos b-pos by auto
 have summable: summable $(\lambda k. bb\ k / aa\ k)$
 proof -
 define $c0$ where $c0 \equiv (aa\ t / bb\ t) \text{ powr } (1 / (1 + \varepsilon)) - t$

```

have ab-k:(aa k / bb k) powr(1/(1+ε)) ≥ k + c0 when k ≥ t for k
  using that
proof (induct k)
  case 0
  then show ?case unfolding c0-def by simp
next
  case (Suc k)
  have ?case when ¬ t ≤ k
  proof -
    have t = Suc k using that Suc.prem by linarith
    then show ?thesis unfolding c0-def by auto
  qed
  moreover have ?case when t ≤ k
  proof -
    have (aa(k+1)/bb(k+1)) powr(1/(1+ε))
      ≥ (aa k / bb k) powr(1/(1+ε)) + 1
      using ratio-large[rule-format, OF that] by blast
    then show ?thesis using Suc(1)[OF that] by simp
  qed
  ultimately show ?case by auto
qed
have summable (λk. 1 / (k + c0) powr (1+ε))
proof -
  have c0 + t > 0 unfolding c0-def
  using aa-bb-pos[of t] by (simp, linarith)
  then have summable (λk. 1 / (k + (c0+t)) powr (1+ε))
  using summable-hurwitz-zeta-real[of 1+ε c0+t]
  apply (subst (asm) powr-minus-divide)
  using ⟨ε > 0⟩ by auto
  then show ?thesis
  apply (rule-tac summable-offset[of - t])
  by (auto simp: algebra-simps)
qed
moreover have bb k / aa k ≤ 1 / (k + c0) powr (1+ε) when k ≥ t for k
proof -
  have (aa t / bb t) powr (1 / (1 + ε)) > 0
  apply simp
  by (metis ⟨∧k. 0 < aa k⟩ ⟨∧k. 0 < bb k⟩ less-numeral-extra(3))
  then have k + c0 > 0 unfolding c0-def using that by linarith
  then have aa k / bb k ≥ (k + c0) powr (1+ε)
  using ab-k[OF that]
  apply (subst (asm) powr-less-eq-inverse-iff')
  using ⟨ε > 0⟩ by auto
  then have inverse (aa k / bb k) ≤ inverse ((k + c0) powr (1+ε))
  apply (elim le-imp-inverse-le)
  using ⟨k + c0 > 0⟩ by auto
  then show ?thesis by (simp add: inverse-eq-divide)
qed
ultimately show ?thesis

```

```

apply (elim summable-comparison-test'[where N=t])
using aa-bb-pos by (simp add: less-eq-real-def)
qed

have  $\exists \infty k. 1 / (M \text{ powr } (\varepsilon / (1 + \varepsilon)) * (\prod_{i=0..k} aa\ i) \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))) > (bb\ (k+1) / aa\ (k+1)) \text{ powr } (\varepsilon / (1 + \varepsilon))$ 
when  $M > 0$  for  $M$ 
proof -
define tt where  $tt \equiv (\lambda i. \text{prod } aa\ \{0..i\} \text{ powr } (2 + 2 / \varepsilon + \delta))$ 
have tt-pos:  $tt\ i > 0$  for  $i$ 
unfolding tt-def
apply (subst powr-gt-zero, induct i)
subgoal by (metis aa-bb-pos(1) order-less-irrefl prod-pos)
subgoal by (metis  $\langle \wedge k. 0 < aa\ k \rangle$  order-less-irrefl prod-pos)
done
have  $\exists \infty i. M < (aa\ (i + 1) / tt\ i * (1 / bb\ (i + 1)))$ 
using limsup-infinity-imp-Inf-many[OF limsup-infi, rule-format, of M]
unfolding tt-def by auto
then have  $\exists \infty i. 1 / (M * tt\ i) > (bb\ (i+1) / aa\ (i+1))$ 
apply (elim INFM-mono)
using  $\langle M > 0 \rangle$  tt-pos by (auto simp: divide-simps algebra-simps)
then have  $\exists \infty i. (1 / (M * tt\ i)) \text{ powr } (\varepsilon / (1 + \varepsilon)) > (bb\ (i+1) / aa\ (i+1)) \text{ powr } (\varepsilon / (1 + \varepsilon))$ 
apply (elim INFM-mono powr-less-mono2[rotated 2])
by (simp-all add: assms(6) pos-add-strict less-eq-real-def)
moreover have  $tt\ i \text{ powr } (\varepsilon / (1 + \varepsilon)) = \text{prod } aa\ \{0..i\} \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
for  $i$ 
unfolding tt-def
apply (auto simp: powr-powr)
using  $\langle \varepsilon > 0 \rangle$  by (simp add: divide-simps, simp add: algebra-simps)
ultimately show ?thesis
apply (elim INFM-mono)
by (smt nonzero-mult-div-cancel-left powr-divide powr-mult powr-one-eq-one
that tt-pos zero-less-divide-iff)
qed

have  $\delta : \forall_F k \text{ in sequentially. } \forall s. ((aa\ (k+s) / bb\ (k+s))) \geq (((aa\ k / bb\ k) \text{ powr } (1 / (1 + \varepsilon))) + s) \text{ powr } (1 + \varepsilon)$ 
using eventually-ge-at-top[of t]
proof eventually-elim
case (elim k)
define ff where  $ff \equiv (\lambda k. (aa\ k / bb\ k) \text{ powr } (1 / (1 + \varepsilon)))$ 
have  $11 : ff\ k+s \leq ff\ (k+s)$  for  $s$ 
proof (induct s)
case 0
then show ?case unfolding ff-def by auto
next
case (Suc s)

```

```

then have ff k + Suc s ≤ ff (k+Suc s)
  using ratio-large[rule-format,of k+s] ‹ t ≤ k ‹ unfolding ff-def by auto
then show ?case by simp
qed
then have (ff k+s) powr (1+ε) ≤ ff (k+s) powr (1+ε) for s
  apply (rule powr-mono2[of 1+ε,rotated 2])
  unfolding ff-def using ‹ε>0 ‹ by auto
then show ?case unfolding ff-def using ‹ε>0 ‹
  apply (auto simp add:powr-powr)
  by (simp add: a-pos aa-def b-pos bb-def le-less)
qed

have 9: (∑ r. 1/((z+real r)powr(1+ε))) ≤ 1/(ε *(z-1) powr ε)
  summable (λi. 1/((z+ real i)powr(1+ε)))
  when z>1 for z
proof -
  define f where f ≡ (λr. 1/((z+ r)powr(1+ε)))
  have ((λx. f (x - 1)) has-integral ((z-1) powr - ε) / ε) {0..}
  proof -
  have ((λx. (z-1 + x) powr (- 1 - ε)) has-integral ((z-1) powr - ε) / ε)
{0<..}
  using powr-has-integral-at-top[of 0 z-1 - 1 - ε,simplified] ‹z>1 ‹ ‹ε>0 ‹
  by auto
  then have ((λx. (z-1 + x) powr (- 1 - ε)) has-integral ((z-1) powr - ε)
/ ε) {0..}
  apply (subst (asm) has-integral-closure[symmetric])
  by (auto simp add: negligible-convex-frontier)
  then show ?thesis
  apply (rule has-integral-cong[THEN iffD1, rotated 1])
  unfolding f-def by (smt powr-minus-divide)
qed
moreover have ∧x. 0 ≤ x ⇒ 0 ≤ f (x - 1) unfolding f-def by simp
moreover have ∧x y. 0 ≤ x ⇒ x ≤ y ⇒ f (y - 1) ≤ f (x - 1) unfolding
f-def
  by (smt assms(6) frac-le powr-mono2 powr-nonneg-iff that)
  ultimately have summable (λi. f (real i)) (∑ i. f (real i)) ≤ (z - 1) powr -
ε / ε
  using decreasing-sum-le-integral[of λx. f (x-1) ((z-1) powr - ε) / ε,simplified]
  by auto
  moreover have (z - 1) powr - ε / ε = 1/(ε *(z-1) powr ε)
  by (simp add: powr-minus-divide)
  ultimately show (∑ i. f (real i)) ≤ 1/(ε *(z-1) powr ε) by auto
  show summable (λi. f (real i)) using ‹summable (λi. f (real i)) ‹ .
qed

have u:(λk.( aa k / bb k)) —————> ∞
proof -
  define ff where ff≡(λx. ereal (aa x / bb x))
  have limsup ff = ∞

```

```

proof -
  define tt where tt ≡ (λi. prod aa {0..i} powr (2 + 2 / ε + δ))
  have tt i ≥ 1 for i
    unfolding tt-def
    apply (rule ge-one-powr-ge-zero)
  subgoal
    apply (rule linordered-nonzero-semiring-class.prod-ge-1)
    by (simp add: a-pos aa-def int-one-le-iff-zero-less)
  subgoal by (simp add: ⟨ε>0⟩ ⟨δ>0⟩ less-imp-le)
  done
  then have limsup (λx. (aa (x + 1) / tt x * (1 / bb (x + 1))))
    ≤ limsup (λx. aa (x + 1) / bb (x + 1))
  apply (intro Limsup-mono eventuallyI)
  apply (auto simp add:field-simps order.order-iff-strict)
  by (metis aa-bb-pos(1) div-by-1 frac-less2 less-irrefl less-numeral-extra(1)
    not-le)
  also have ... = limsup (λx. aa x / bb x)
    by (subst limsup-shift,simp)
  finally have limsup (λx. ereal (aa (x + 1) / tt x * (1 / bb (x + 1))))
    ≤ limsup (λx. ereal (aa x / bb x)) .
  moreover have limsup (λx. ereal (aa (x + 1) / tt x * (1 / bb (x + 1))))
    = ∞ using limsup-infi unfolding tt-def by auto
  ultimately show ?thesis
    unfolding ff-def using ereal-infty-less-eq2(1) by blast
qed
then have limsup (λk. ff (k+t)) = ∞
  by (simp add: limsup-shift-k)
moreover have incseq (λk. ff (k+t))
proof (rule incseq-SucI)
  fix k::nat
  define gg where gg≡(λx. (aa x / bb x))
  have (gg (k+t)) powr (1 / (1 + ε)) + 1
    ≤ (gg (Suc (k+t))) powr (1 / (1 + ε))
    using ratio-large[rule-format, of k+t,simplified] unfolding gg-def
    by auto
  then have (gg (k+t)) powr (1 / (1 + ε))
    ≤ (gg (Suc (k+t))) powr (1 / (1 + ε))
    by auto
  then have gg (k+t) ≤ gg (Suc (k+t))
    by (smt aa-bb-pos(1) aa-bb-pos(2) assms(6) divide-pos-pos gg-def
      powr-less-mono2)
  then show ff (k + t) ≤ ff (Suc k + t)
    unfolding gg-def ff-def by auto
qed
ultimately have (λk. ff (k+t)) ⟶ ∞ using incseq-tendsto-limsup
  by fastforce
then show ?thesis unfolding ff-def
  unfolding tendsto-def
  apply (subst eventually-sequentially-seg[symmetric,of - t])

```

```

    by simp
  qed

  define  $\xi$  where  $\xi = \text{suminf } (\lambda k. \text{bb } k / \text{aa } k)$ 
  have  $10: \forall F k \text{ in sequentially. } |\xi - (\sum k = 0..k. \text{bb } k / \text{aa } k)|$ 
    <  $2 / \varepsilon * (\text{bb } (k+1) / \text{aa } (k+1)) \text{ powr } (\varepsilon / (1+\varepsilon))$ 
    using  $\delta[\text{THEN sequentially-offset, of } 1] \text{ eventually-ge-at-top[of } t]$ 
      u[unfolded tendsto-PIfty, rule-format, THEN sequentially-offset
        , of  $(2 \text{ powr } (1/\varepsilon) / (2 \text{ powr } (1/\varepsilon) - 1)) \text{ powr } (1+\varepsilon) 1]$ 
  proof eventually-elim
    case (elim k)
    define tt where  $tt \equiv (\text{aa } (k + 1) / \text{bb } (k + 1)) \text{ powr } (1 / (1 + \varepsilon))$ 
    have  $tt > 1$ 
    proof -
      have  $(\text{aa } k / \text{bb } k) \text{ powr } (1 / (1 + \varepsilon)) > 0$ 
      by (metis a-pos aa-def b-pos bb-def divide-eq-0-iff less-irrefl
        of-int-eq-0-iff powr-gt-zero)
      then show ?thesis using ratio-large[rule-format, OF  $\langle k \geq t \rangle$ ] unfolding tt-def
    by argo
  qed
  have  $|\xi - (\sum i = 0..k. \text{bb } i / \text{aa } i)| = |\sum i. \text{bb } (i+(k+1)) / \text{aa } (i+(k+1))|$ 
    unfolding  $\xi$ -def
    apply (subst suminf-minus-initial-segment[of - k+1, OF summable])
    using atLeast0AtMost lessThan-Suc-atMost by auto
  also have  $\dots = (\sum i. \text{bb } (i+(k+1)) / \text{aa } (i+(k+1)))$ 
  proof -
    have  $(\sum i. \text{bb } (i+(k+1)) / \text{aa } (i+(k+1))) > 0$ 
    apply (rule suminf-pos)
    subgoal using summable[THEN summable-ignore-initial-segment, of k+1] .
    subgoal by (simp add: a-pos aa-def b-pos bb-def)
    done
    then show ?thesis by auto
  qed
  also have  $\dots \leq (\sum i. 1 / (tt + i) \text{ powr } (1 + \varepsilon))$ 
  proof (rule suminf-le)
    define ff where  $ff \equiv (\lambda k n. (\text{aa } (k+1) / \text{bb } (k+1)) \text{ powr } (1 / (1 + \varepsilon)) +$ 
    real n)
    have  $\text{bb } (n + (k + 1)) / \text{aa } (n + (k + 1)) \leq 1 / (\text{ff } k n) \text{ powr } (1 + \varepsilon)$  for n
    proof -
      have  $\text{ff } k n \text{ powr } (1 + \varepsilon) \leq \text{aa } (k + n + 1) / \text{bb } (k + n + 1)$ 
      using elim(1)[rule-format, of n] unfolding ff-def by auto
      moreover have  $\text{ff } k n \text{ powr } (1 + \varepsilon) > 0$ 
      unfolding ff-def by (smt elim(2) of-nat-0-le-iff powr-gt-zero ratio-large)
      ultimately have  $1 / \text{ff } k n \text{ powr } (1 + \varepsilon) \geq \text{bb } (k + (n + 1)) / \text{aa } (k +$ 
      (n + 1))
      apply (drule-tac le-imp-inverse-le)
      by (auto simp add: inverse-eq-divide)
      then show ?thesis by (auto simp: algebra-simps)
    qed
  qed

```



```

then show  $\bigwedge n. bb (n + (k + 1)) / aa (n + (k + 1)) \leq 1 / (tt + real n)$ 
powr (1 +  $\varepsilon$ )
  unfolding ff-def tt-def by auto
  show summable ( $\lambda i. bb (i + (k + 1)) / aa (i + (k + 1))$ )
    using summable[THEN summable-ignore-initial-segment, of k+1] .
  show summable ( $\lambda x. 1 / (tt + real x) \text{ powr } (1 + \varepsilon)$ )
    using g(2)[OF <tt>1] .
qed
also have  $\dots \leq 1 / (\varepsilon * (tt - 1) \text{ powr } \varepsilon)$ 
  using g[OF <tt>1] by simp
also have  $\dots < 2 / (\varepsilon * tt \text{ powr } \varepsilon)$ 
proof -
  have  $((2 \text{ powr } (1/\varepsilon) / (2 \text{ powr } (1/\varepsilon) - 1)) \text{ powr } (1 + \varepsilon)) < (aa (k+1) / bb$ 
( $k+1$ ))
    using elim(3) by auto
  then have  $2 \text{ powr } (1/\varepsilon) / (2 \text{ powr } (1/\varepsilon) - 1) < tt$ 
    unfolding tt-def
    apply (drule-tac powr-less-mono2[rotated 2, where a=1 / (1 +  $\varepsilon$ )])
    using  $\langle \varepsilon > 0 \rangle$  apply (auto simp: powr-powr)
    by (subst (asm) powr-one, auto simp add: field-simps)
  then have  $tt < (tt - 1) * (2 \text{ powr } (1/\varepsilon))$ 
    using  $\langle \varepsilon > 0 \rangle$  by (auto simp: divide-simps algebra-simps)
  then have  $tt \text{ powr } \varepsilon < 2 * (tt - 1) \text{ powr } \varepsilon$ 
    apply (drule-tac powr-less-mono2[rotated 2, where a= $\varepsilon$ ])
    using  $\langle \varepsilon > 0 \rangle$   $\langle tt > 1 \rangle$  by (auto simp: powr-powr powr-mult)
  then show ?thesis
    using  $\langle \varepsilon > 0 \rangle$   $\langle tt > 1 \rangle$  by (auto simp: divide-simps)
qed
also have  $\dots = 2 / \varepsilon * (bb (k + 1) / aa (k + 1)) \text{ powr } (\varepsilon / (1 + \varepsilon))$ 
  unfolding tt-def
  using  $\langle \varepsilon > 0 \rangle$ 
  by (auto simp: powr-powr divide-simps algebra-simps powr-divide less-imp-le)
finally show ?case .
qed

define pq where  $pq \equiv (\lambda k. \text{quotient-of } (\sum i=0..k. \text{of-int } (b i) / \text{of-int } (a i)))$ 
define p q where  $p \equiv \text{fst } \circ pq$  and  $q \equiv \text{snd } \circ pq$ 
have coprime-pq: coprime ( $p k$ ) ( $q k$ )
  and q-pos: q k > 0 and pq-sum: p k / q k =  $(\sum i=0..k. b i / a i)$  for  $k$ 
proof -
  have eq: quotient-of  $(\sum i=0..k. \text{of-int } (b i) / \text{of-int } (a i)) = (p k, q k)$ 
    by (simp add: p-def q-def pq-def)
  from quotient-of-coprime[OF eq] show coprime ( $p k$ ) ( $q k$ ) .
  from quotient-of-denom-pos[OF eq] show  $q k > 0$  .
  have  $(\sum i=0..k. b i / a i) = \text{of-rat } (\sum i=0..k. \text{of-int } (b i) / \text{of-int } (a i))$ 
    by (simp add: of-rat-sum of-rat-divide)
  also have  $(\sum i=0..k. \text{rat-of-int } (b i) / \text{rat-of-int } (a i)) =$ 
 $\text{rat-of-int } (p k) / \text{rat-of-int } (q k)$ 
    using quotient-of-div[OF eq] by simp

```

```

finally show  $p\ k / q\ k = (\sum\ i=0..k.\ b\ i / a\ i)$  by (simp add:of-rat-divide)
qed

have  $\xi$ -Inf-many: $\exists\ \infty\ k.\ |\xi - p\ k / q\ k| < 1 / q\ k\ \text{powr}\ (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
proof -
  have *: $\exists\ \infty\ k.\ (bb\ (Suc\ k) / aa\ (Suc\ k))\ \text{powr}\ (\varepsilon / (1 + \varepsilon))$ 
     $< \varepsilon / (2 * \text{prod}\ aa\ \{0..k\}\ \text{powr}\ (2 + \delta * \varepsilon / (1 + \varepsilon)))$ 
    using  $\gamma$ [of  $(2 / \varepsilon)\ \text{powr}\ ((1+\varepsilon)/\varepsilon)$ ]  $\langle\varepsilon>0\rangle$ 
    by (auto simp:powr-powr)
  have **: $\forall\ \infty\ k.\ |\xi - (\sum\ k = 0..k.\ bb\ k / aa\ k)|$ 
     $< 2 / \varepsilon * (bb\ (k + 1) / aa\ (k + 1))\ \text{powr}\ (\varepsilon / (1 + \varepsilon))$ 
    using  $10$ [unfolded cofinite-eq-sequentially[symmetric]] .
  from INFM-conjI[OF * **] show ?thesis
proof (elim INFM-mono)
    fix  $k$  assume asm: $(bb\ (Suc\ k) / aa\ (Suc\ k))\ \text{powr}\ (\varepsilon / (1 + \varepsilon))$ 
       $< \varepsilon / (2 * \text{prod}\ aa\ \{0..k\}\ \text{powr}\ (2 + \delta * \varepsilon / (1 + \varepsilon))) \wedge$ 
       $|\xi - (\sum\ k = 0..k.\ bb\ k / aa\ k)|$ 
       $< 2 / \varepsilon * (bb\ (k + 1) / aa\ (k + 1))\ \text{powr}\ (\varepsilon / (1 + \varepsilon))$ 
    have  $|\xi - \text{real-of-int}\ (p\ k) / \text{real-of-int}\ (q\ k)|$ 
       $= |\xi - (\sum\ k = 0..k.\ bb\ k / aa\ k)|$ 
      using pq-sum unfolding aa-def bb-def by auto
    also have ...  $< (2 / \varepsilon) * (bb\ (k+1) / aa\ (k+1))\ \text{powr}\ (\varepsilon / (1+\varepsilon))$ 
      using asm by auto
    also have ...  $< (2 / \varepsilon) * (\varepsilon / (2 * \text{prod}\ aa\ \{0..k\}\ \text{powr}\ (2 + \delta * \varepsilon / (1 + \varepsilon))))$ 
    apply (rule mult-strict-left-mono)
    using asm  $\langle\varepsilon>0\rangle$  by auto
    also have ...  $= 1 / \text{prod}\ aa\ \{0..k\}\ \text{powr}\ (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
      using  $\langle\varepsilon>0\rangle$  by auto
    also have ...  $\leq 1 / q\ k\ \text{powr}\ (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
    proof -
      have  $q\ k \leq \text{prod}\ aa\ \{0..k\}$ 
      proof (induct k)
        case  $0$ 
        then show ?case unfolding q-def pq-def aa-def
          apply (simp add:rat-divide-code of-int-rat quotient-of-Fract)
          using aa-bb-pos[of 0,unfolded aa-def bb-def] unfolding Let-def normal-
ize-def
          apply auto
          by (metis div-by-1 less-imp-le less-trans nonneg1-imp-zdiv-pos-iff
            not-less zdiv-mono2)
        next
        case  $(Suc\ k)$ 
        define de where  $de \equiv \text{snd} \circ \text{quotient-of}$ 
        have real-of-int  $(q\ (Suc\ k))$ 
           $= de\ (\sum\ i=0..Suc\ k.\ \text{of-int}\ (b\ i) / \text{of-int}\ (a\ i))$ 
          unfolding q-def pq-def de-def by simp
        also have ...  $= de\ ((\sum\ i=0..k.\ \text{of-int}\ (b\ i) / \text{of-int}\ (a\ i))$ 
           $+ \text{of-int}\ (b\ (Suc\ k)) / \text{of-int}\ (a\ (Suc\ k)))$ 

```

```

    by simp
  also have ... ≤ de (∑ i=0..k. of-int (b i) / of-int (a i))
    * de (of-int (b (Suc k)) / of-int (a (Suc k)))
    using snd-quotient-plus-leq[folded de-def] by presburger
  also have ... = q k * de (of-int (b (Suc k)) / of-int (a (Suc k)))
    unfolding q-def pq-def de-def by auto
  also have ... = q k * snd (Rat.normalize (b (Suc k), a (Suc k)))
    by (simp add:rat-divide-code of-int-rat quotient-of-Fract de-def)
  also have ... ≤ q k * aa (Suc k)
    using aa-bb-pos[of Suc k] q-pos[of k]
    unfolding normalize-def aa-def bb-def Let-def
    apply auto
    by (metis div-by-1 int-one-le-iff-zero-less less-trans
      nonneg1-imp-zdiv-pos-iff not-less zdiv-mono2 zero-less-one)
  also have ... ≤ prod aa {0..k} * aa (Suc k)
    using Suc aa-bb-pos[of Suc k] by auto
  also have ... = prod aa {0..Suc k}
    by (simp add: prod.atLeast0-atMost-Suc)
  finally show ?case .
qed
then show ?thesis
  apply (rule-tac divide-left-mono)
  apply (rule powr-mono2)
  using ‹δ>0› ‹ε>0› q-pos[of k]
  apply (auto simp:powr-mult[symmetric])
  by (metis aa-bb-pos(1) less-irrefl)
qed
finally show |ξ - p k / q k| < 1 / q k powr (2 + δ * ε / (1 + ε)) .
qed
qed

```

define pqs **where** pqs ≡ {(p, q). q>0 ∧ coprime p q ∧ |ξ - real-of-int p / real-of-int q| < 1 / q powr (2 + δ * ε / (1 + ε))}

have ξ-infinite:infinite pqs

proof -

define A **where** A ≡ {k. |ξ - (p k) / (q k)| < 1 / (q k) powr (2 + δ * ε / (1 + ε))}

note ξ-Inf-many

then have infinite A

unfolding Inf-many-def A-def **by** auto

moreover have inj-on (λk. (p k, q k)) A

proof -

define g **where** g ≡ (λi. rat-of-int (b i) / rat-of-int (a i))

define f **where** f ≡ (λk. ∑ i = 0..k. g i)

have g-pos:g i>0 **for** i

unfolding g-def **by** (simp add: a-pos b-pos)

have strict-mono f **unfolding** strict-mono-def f-def

proof safe

```

fix x y::nat assume x < y
then have sum g {0..y} - sum g {0..x} = sum g {x<..y}
  apply (subst Groups-Big.sum-diff[symmetric])
  by (auto intro:arg-cong2[where f=sum])
also have ... > 0
  apply (rule ordered-comm-monoid-add-class.sum-pos)
  using ⟨x<y⟩ g-pos by auto
finally have sum g {0..y} - sum g {0..x} > 0 .
then show sum g {0..x} < sum g {0..y} by auto
qed
then have inj f using strict-mono-imp-inj-on by auto
then have inj (quotient-of o f) by (simp add: inj-compose quotient-of-inj)
then have inj (λk. (p k, q k))
  unfolding f-def p-def q-def pq-def comp-def
  apply (fold g-def)
  by auto
then show ?thesis by (auto elim:subset-inj-on)
qed
moreover have (λk. (p k, q k)) ‘ A ⊆ pqs
  unfolding A-def pqs-def using coprime-pq q-pos by auto
ultimately show ?thesis
  apply (elim infinite-inj-imageE)
  by auto
qed
moreover have finite pqs if ξ ∈ ℚ
proof -
  obtain m n where ξ-mn:ξ = (of-int m / of-int n) and coprime m n n>0
  proof -
    obtain m n where mn:|ξ| = (of-nat m / of-nat n) coprime m n n≠0
      using Rats-abs-nat-div-natE[OF ⟨ξ ∈ ℚ⟩ Rats-abs-nat-div-natE]
      by metis
    define m' and n':int
      where m'=(if ξ > 0 then nat m else -nat m) and n'=nat n
    then have ξ = (of-int m' / of-int n') coprime m' n' n'>0
      using mn by auto
    then show ?thesis using that by auto
  qed
  have pqs ⊆ {(m,n)} ∪ {x. x ∈ pqs ∧ - |m| - 1 ≤ fst x ∧ fst x ≤ |m| + 1 ∧
0 < snd x ∧ snd x < n }
  proof (rule subsetI)
    fix x assume x ∈ pqs
    define p q where p ≡ fst x and q=snd x
    have q>0 coprime p q and pq-less:|ξ - p / q|
      < 1 / q powr (2 + δ * ε / (1 + ε))
      using ⟨x∈pqs⟩ unfolding p-def q-def pqs-def by auto
    have q<n:q<n when m≠p ∨ n≠q
    proof -
      have m * q ≠ n * p using that ⟨coprime m n⟩ ⟨coprime p q⟩ ⟨q>0⟩ ⟨n>0⟩
      by (metis eq-rat(1) fst-conv int-one-le-iff-zero-less mult.commute normal-

```

ize-stable

```

      not-one-le-zero quotient-of-Fract snd-conv)
then have  $1/(n*q) \leq |m/n - p/q|$ 
  using  $\langle q>0 \rangle \langle n>0 \rangle$ 
  apply (auto simp:field-simps)
by (metis add-diff-cancel-left' diff-diff-eq2 diff-zero less-irrefl not-le of-int-diff

      of-int-lessD of-int-mult)
also have ...  $< 1 / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
  using pq-less unfolding  $\xi\text{-mn}$  by auto
also have ...  $\leq 1 / q^{\wedge}2$ 
proof -
  have real-of-int ( $q^2$ ) =  $q \text{ powr } 2$ 
  apply (subst powr-numeral)
  unfolding power2-eq-square using  $\langle q>0 \rangle$  by auto
also have ...  $\leq q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
  apply (rule powr-mono)
  using  $\langle q>0 \rangle \langle \delta>0 \rangle \langle \varepsilon>0 \rangle$  by auto
  finally have real-of-int ( $q^2$ )
     $\leq \text{real-of-int } q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$  .
  moreover have real-of-int  $q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon)) > 0$  using  $\langle 0 <$ 
 $q \rangle$  by auto
  ultimately show ?thesis by (auto simp:field-simps)
qed
finally have  $1 / (n * q) < 1 / q^2$  .
then show ?thesis using  $\langle q>0 \rangle \langle n>0 \rangle$ 
  unfolding power2-eq-square by (auto simp:field-simps)
qed
moreover have  $- |m| - 1 \leq p \wedge p \leq |m| + 1$  when  $m \neq p \vee n \neq q$ 
proof -
  define  $qn$  where  $qn \equiv q/n$ 
  have  $0 < qn \wedge qn < 1$  unfolding qn-def using q-lt-n[OF  $\langle m \neq p \vee n \neq q \rangle$ ]  $\langle q>0 \rangle$ 
by auto

  have  $|m/n - p/q| < 1 / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
  using pq-less unfolding  $\xi\text{-mn}$  by simp
then have  $|p/q - m/n| < 1 / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$  by simp
then have  $m/n - 1 / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
   $< p/q \wedge p/q < m/n + 1 / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
  unfolding abs-diff-less-iff by auto
then have  $qn*m - q / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon)) < p$ 
   $\wedge p < qn*m + q / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
  unfolding qn-def using  $\langle q>0 \rangle$  by (auto simp:field-simps)
moreover have  $- |m| - 1 \leq qn*m - q / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
proof -
  have  $- |m| \leq qn*m$  using  $\langle 0 < qn \rangle \langle qn < 1 \rangle$ 
  apply (simp add: abs-if)
  by (smt (verit, best) mult-nonneg-nonneg of-int-nonneg)
moreover have  $- 1 \leq - q / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 

```

```

proof –
  have  $q = q \text{ powr } 1$  using  $\langle 0 < q \rangle$  by auto
  also have  $\dots \leq q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
    apply (rule powr-mono)
    using  $\langle q > 0 \rangle \langle \delta > 0 \rangle \langle \varepsilon > 0 \rangle$  by auto
  finally have  $q \leq q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$  .
  then show ?thesis using  $\langle 0 < q \rangle$  by auto
qed
ultimately show ?thesis by auto
qed
moreover have  $qn * m + q / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon)) \leq |m| + 1$ 
proof –
  have  $qn * m \leq |m|$  using  $\langle 0 < qn \rangle \langle qn < 1 \rangle$ 
  apply (simp add: abs-if mult-left-le-one-le)
  by (meson less-eq-real-def mult-pos-neg neg-0-less-iff-less of-int-less-0-iff
order-trans)
  moreover have  $q / q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon)) \leq 1$ 
proof –
  have  $q = q \text{ powr } 1$  using  $\langle 0 < q \rangle$  by auto
  also have  $\dots \leq q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$ 
    apply (rule powr-mono)
    using  $\langle q > 0 \rangle \langle \delta > 0 \rangle \langle \varepsilon > 0 \rangle$  by auto
  finally have  $q \leq q \text{ powr } (2 + \delta * \varepsilon / (1 + \varepsilon))$  .
  then show ?thesis using  $\langle 0 < q \rangle$  by auto
qed
ultimately show ?thesis by auto
qed
ultimately show ?thesis by auto
qed
ultimately show  $x \in \{(m, n)\} \cup \{x \in pqs. - |m| - 1 \leq fst\ x \wedge fst\ x \leq |m|$ 
+ 1
       $\wedge 0 < snd\ x \wedge snd\ x < n\}$ 
    using  $\langle x \in pqs \rangle \langle q > 0 \rangle$  unfolding p-def q-def by force
qed
moreover have finite  $\{x. x \in pqs \wedge - |m| - 1 \leq fst\ x \wedge fst\ x \leq |m| + 1 \wedge$ 
0 < snd  $x \wedge snd\ x < n\}$ 
proof –
  have finite  $(\{- |m| - 1 .. |m| + 1\} \times \{0 < .. < n\})$  by blast
  moreover have  $\{x. x \in pqs \wedge - |m| - 1 \leq fst\ x \wedge fst\ x \leq |m| + 1 \wedge 0 < snd$ 
x  $\wedge snd\ x < n\} \subseteq$ 
     $(\{- |m| - 1 .. |m| + 1\} \times \{0 < .. < n\})$ 
  by auto
  ultimately show ?thesis
  using finite-subset by blast
qed
ultimately show ?thesis using finite-subset by auto
qed
ultimately show ?thesis
  apply (fold  $\xi$ -def)

```

```

using RothsTheorem[rule-format,of  $\xi 2 + \delta * \varepsilon / (1 + \varepsilon)$ ,folded pqs-def]
   $\langle \delta > 0 \rangle \langle \varepsilon > 0 \rangle$ 
apply (auto simp:divide-simps )
by (meson mult-le-0-iff not-less)
qed

```

1.3 Acknowledgements

A.K.-A. and W.L. were supported by the ERC Advanced Grant ALEXANDRIA (Project 742178) funded by the European Research Council and led by Professor Lawrence Paulson at the University of Cambridge, UK. Thanks to Iosif Pinelis for his clarification on MathOverflow regarding the summability of the series in *RothsTheorem.HanclRucki2* <https://mathoverflow.net/questions/323069/why-is-this-series-summable> and to Manuel Eberl for his helpful comments.

end

References

- [1] J. Hančl and P. Rucki. The transcendence of certain infinite series. *Rocky Mountain Journal of Mathematics*, 35(2):531–537, 2005.
- [2] K. F. Roth. Rational approximations to algebraic numbers. *Mathematika*, 2(3):1–20, 1955.